

姚攀的博客 1.01^365=31.78

Engineers are versatile minds who create links between science, technology, and society.

- 目录视图
- 摘要视图
- RSS 订阅

«从Lucene到Elasticsearch：全文检索实战»

希望本书能对您的工作和学习带来帮助，感谢支持！



当当 京东 亚马逊 天猫

Lucene、ES、ELK开发交流群:370734940

 加入QQ群


个人资料



yyyseb

关注

发私信



访问： 844552次

积分： 8642

等级： 

排名： 第2540名

图灵赠书——程序员11月书单 【思考】Python这么厉害的原因竟然是！ 感恩节赠书：《深度学习》等异步社区优秀图书和作译者评选启动！ 每周荐书：京东架构、Linux内核、Python全栈

Spring全家桶(五)Bean的多种配置方法

标签： spring bean 工厂方法

2017-05-16 11:29 732人阅读 评论(0)

分类： spring (8)

版权声明：本文为博主原创文章,未经博主允许禁止转载(<http://blog.csdn.net/napoay>)

目录(?) [+]

十一、工厂方法配置Bean

通过全类名方法配置Bean底层采用的是反射，除此之外还可以通过工厂方法(静态工厂方法&实例工厂方法)、FactoryBean来配置Bean。

静态工厂方法创建Bean是将对象创建的过程封装到静态方法中，当客户端需要对象时，只需要简单地调用静态方法而不需要关系创建对象的细节。
要声明通过静态方法调用Bean，需要在Bean的class属性里指定该工厂的方法的类，同时在factory-method属性中指定工厂方法的名称，最后使用 <constructor-arg> 元素为该方法传递方法参数。

11.1静态工厂方法

Car类：

```
1 package com.stuspring.factory;
2
3 /**
4  * Created by bee on 17/4/28.
5  */
6 public class Car {
7     private String brand;
8     private double price;
9
10
11     public void setBrand(String brand) {
12         this.brand = brand;
```

原创： 202篇 转载： 2篇

译文： 6篇 评论： 451条

我的课程

Elasticsearch 5.4新闻搜索项目实战

StackOverFlow

<http://stackoverflow.com/users/6526424>
统计

阅读排行

[搜索]ElasticSearch Java Api(... (61756)

Elasticsearch索引mapping的写... (36806)

Elasticsearch 5.X下JAVA API使... (31047)

搭建Elasticsearch 5.4分布式集群 (29631)

Elasticsearch 5.1.1 head插件安... (25822)

Elasticsearch java api(五) Bulk... (21822)

ElasticSearch Java Api(四)-删... (19887)

mac命令行启动tomcat (18615)

Elasticsearch 5 Ik+pinyin分词... (17900)

Elasticsearch shield权限管理详解 (17369)

博客专栏

Spring全家桶

文章： 10篇
阅读： 10622

MapReduce编程

文章： 7篇
阅读： 24325

Lucene&Elasticsearch&ELK 专栏

文章： 42篇
阅读： 461286

数据库

文章： 9篇
阅读： 18151

J2EE

文章： 24篇
阅读： 92799

ruby on rails

文章： 18篇
阅读： 37381

数据结构与算法

```
13     }
14
15     public void setPrice(double price) {
16         this.price = price;
17     }
18
19     public String getBrand() {
20         return brand;
21     }
22
23     public double getPrice() {
24         return price;
25     }
26
27     public Car() {
28     }
29
30     public Car(String brand, double price) {
31         this.brand = brand;
32         this.price = price;
33     }
34
35     @Override
36     public String toString() {
37         return "Car{" +
38             "brand='" + brand + '\'' +
39             ", price=" + price +
40             '}';
41     }
42 }
```

静态工厂方法：

```
1 package com.stuspring.factory;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 /**
7  * Created by bee on 17/4/28.
8  */
9 public class StaticCarFactory {
10
11
12     private static Map<String,Car> cars=new HashMap<String, Car>();
13
14     static {
15         cars.put("audi",new Car("Audi",300000));
16         cars.put("ford",new Car("Ford",200000));
17     }
18
19
20     //静态工厂方法
21     public static Car getCar(String name){
22         return cars.get(name);
23     }
24 }
```



文章: 13篇
阅读: 17571

文章分类

Elasticsearch (43)

Lucene (15)

hadoop (16)

J2EE (35)

数据结构 (16)

前端 (16)

ruby (18)

机器学习 (4)

python (5)

linux (11)

latex (2)

数据库 (12)

spring (9)

tools (1)

hibernate (1)

mybatis (1)

生活 (1)

文章存档

2017年12月 (9)

2017年11月 (1)

2017年10月 (3)

2017年09月 (2)

2017年08月 (1)

展开

最新评论

MapReduce编程(六) 从HDFS导入数据到Elasticsearch
qq_38094271 : 用最新版本的那个 成功了。
注意json 文件最后一行要换行

《从Lucene到Elasticsearch:全文检索实战...
牙小木 : 书已买, 正在看

Lucene扩展停用词字典与自定义词库
yyyseb : @cuipeng6561:这个文件在ik的目录下。

Lucene扩展停用词字典与自定义词库
cuipeng6561 : IKAnalyzer.cfg.xml 这个文件是要自己写么?

新人千万不要在 Windows 上使用 Ruby on Rails
韩大喵 : 对于新手的我来说的, 相当中肯!

《从Lucene到Elasticsearch:全文检索实战...
艾鹤 : 6666, 恭喜发财。

Elasticsearch 5.X下JAVA API使用指南
qq_41296216 : @napoay:maven坐标是 <groupId>org.elasticsearch.plugin:elasticsearch-analysis-icu</groupId> <version>5.6.2</version> <scope>test</scope></div><div>Elasticsearch java api(五) Bulk批量索引
yyyseb : @wangmaohong0717:可以不指定, 会自动生成。</div><div>Elasticsearch 5.X下JAVA API使用指南
yyyseb : @qq_41296216:贴一下maven坐标吧, 或者, 博客上有QQ群, 加群讨论</div></div></div><div data-bbox="324 34 934 965" data-label="Complex-Block"><div><pre>23 }
24
25 }</pre></div><div><p>配置Bean:</p></div><div><pre>1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4 xsi:schemaLocation="http://www.springframework.org/schema
5
6 <!--通过静态工厂方法配置Bean实例。注意, 不是静态工厂方法实例-->
7
8 <bean id="car1" class="com.stuspring.factory.StaticCarFactor
9 factory-method="getCar">
10 <constructor-arg value="audi" />
11 </bean>
12
13 </beans></pre></div><div><p>测试:</p></div><div><pre>1 package com.stuspring.factory;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 /**
7 * Created by bee on 17/4/28.
8 */
9 public class Main {
10 public static void main(String[] args) {
11 ApplicationContext ctx = new ClassPathXmlApplicationContext("beans-fact
12 Car car1 = (Car) ctx.getBean("car1");
13 System.out.println(car1);
14 }
15 }</pre></div><div><p>运行结果:</p></div><div><pre>1 Car{brand='Audi', price=300000.0}</pre></div><div><h2>11.2 实例工厂方法</h2></div><div><pre>1 package com.stuspring.factory;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 /**
7 * Created by bee on 17/4/28.
8 */</pre></div></div><div data-bbox="43 969 957 981" data-label="Page-Footer"><div>http://blog.csdn.net/napoay/article/details/72238719</div><div>3/7</div></div>

Elasticsearch java api(五) Bulk批量索引
hello-friend : 批量索引数据的时候, 必须指定_id吗

```
9 public class InstanceCarFactory {
10
11     private Map<String,Car> cars=null;
12
13
14     public InstanceCarFactory(){
15         cars=new HashMap<String, Car>();
16         cars.put("audi",new Car("Audi",300000));
17         cars.put("ford",new Car("Ford",400000));
18     }
19
20     public Car getCar(String carname){
21         return cars.get(carname);
22     }
23
24 }
```

配置Bean:

```
1 <!--配置工厂实例-->
2 <bean id="instanceFactory" class="com.stuspring.factory.InstanceCarFactory"
3 <bean id="car2" factory-bean="instanceFactory" factory-method="getCar">
4     <constructor-arg value="ford"/>
5 </bean>
```

测试:

```
1 Car car2= (Car) ctx.getBean("car2");
2 System.out.println(car2);
```

打印结果:

```
1 Car{brand='Ford', price=400000.0}
```

十二、FactoryBean配置Bean

Spring中有2种类型的Bean, 一种是普通Bean, 另一种是工厂Bean, 即FactoryBean。工厂Bean跟普通Bean不同, 其返回的对象不是指定类的一个实例, 而是工厂Bean的getObject方法所返回的对象。

创建一个CarFactoryBean类, 继承FactoryBean接口:

```
1 package com.stuspring.factorybean;
2
3 import org.springframework.beans.factory.FactoryBean;
4
5 /**
6  * Created by bee on 17/4/28.
7  */
8 public class CarFactoryBean implements FactoryBean<Car>{
```

```
9
10     private String brand;
11
12     public void setBrand(String brand) {
13         this.brand = brand;
14     }
15
16     public Car getObject() throws Exception {
17         return new Car(brand,8000000);
18     }
19
20     public Class<?> getObjectType() {
21         return Car.class;
22     }
23
24     public boolean isSingleton() {
25         return false;
26     }
27 }
```

配置Bean:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans http://v
5
6       <bean id="car1" class="com.stuspring.factorybean.CarFactoryBean">
7           <property name="brand" value="Ford"/>
8       </bean>
9       <bean id="car2" class="com.stuspring.factorybean.CarFactoryBean">
10          <property name="brand" value="BMW"/>
11      </bean>
12 </beans>
```

测试:

```
1 package com.stuspring.factorybean;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5
6 /**
7  * Created by bee on 17/4/28.
8  */
9 public class Main {
10
11     public static void main(String[] args) {
12
13         ApplicationContext ctx=new ClassPathXmlApplicationContext("beans-beanf
14         Car car1= (Car) ctx.getBean("car1");
15         System.out.println(car1);
16         Car car1Copy= (Car) ctx.getBean("car1");
17         System.out.println(car1Copy);
```

```
18         System.out.println(car1==car1Copy);
19         Car car2= (Car) ctx.getBean("car2");
20         System.out.println(car2);
21     }
22 }
```

打印结果:

```
1 Car{brand='Ford', price=8000000.0}
2 Car{brand='Ford', price=8000000.0}
3 false
4 Car{brand='BMW', price=8000000.0}
```

13.通过注解配置Bean

13.1基于注解配置Bean

13.2基于注解来装配Bean的属性

顶

0

踩

0

- [上一篇](#) Spring全家桶(四)Bean的生命周期
- [下一篇](#) Spring全家桶(六)必知必会的java注解技术

相关文章推荐

- Spring全家桶(四)Bean的生命周期
 - MySQL在微信支付下的高可用运营--莫晓东
 - 注解的力量 -----Spring 2.5 JPA hibernate 使用方法...
 - 容器技术在58同城的实践--姚远
 - spring管理hibernate的SessionFactory-多种配置方法
 - SDCC 2017之容器技术实战线上峰会
 - spring全家桶+redis+nginx+activeq+solr商城项目
 - SDCC 2017之数据库技术实战线上峰会
- Vuejs全家桶系列（五）---属性和方法
 - 腾讯云容器服务架构实现介绍--董晓杰
 - spring 配置bean的方法及依赖注入发方式
 - 微博热点事件背后的数据库运维心得--张冬洪
 - Spring_11_通过工厂方法配置 Bean
 - Spring框架学习（8）：通过工厂方法配置bean
 - Spring通过工厂方法配置Bean
 - Spring-基于注解的配置I03Bean作用范围和生命周...

查看评论

暂无评论

您还没有登录,请[登录](#)或[注册](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 